Enterprise Computing Series

WinFS: File System And Security Drilldown

Sanjay Anand, GPM WinFS FS
Sameet Agarwal, Dev Mgr. WinFS FS

March 23rd, 2003

Microsoft

Agenda

- “Hello!!”

- “Marriage”

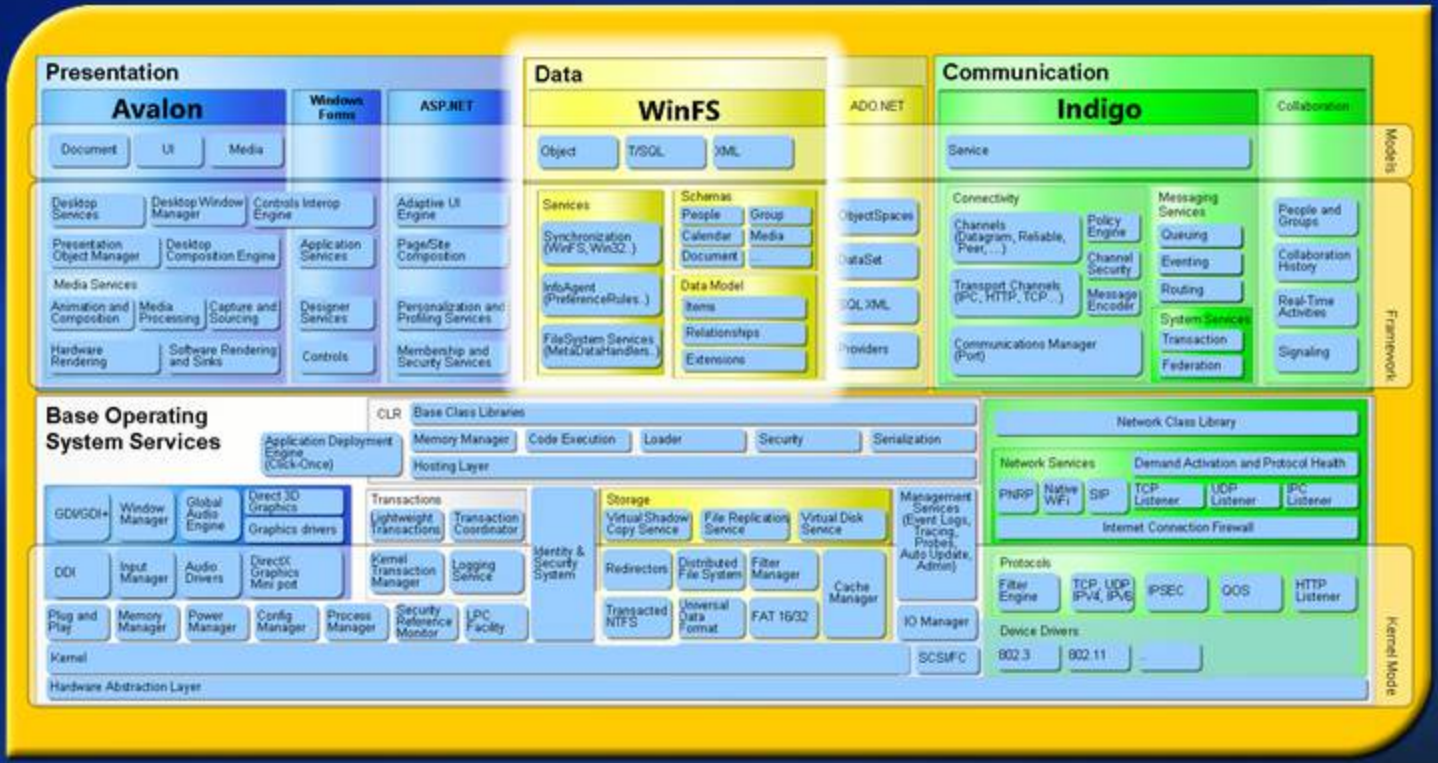
- “Freedom”

- “Safety”

“Hello!!”

A small primer on WinFS…

Longhorn architecture



Windows File System is . . .

WinFS is the active storage subsystem in Longhorn for searching, organizing, and sharing data

- All end-user data lives in Longhorn
- New user experience in Longhorn Shell
- A trustworthy place to store data
- Data model built on relational database technology
- Filesystem capabilities built on NTFS
- Everyday Information - domain-specific schemas
- Services that make data active

User Benefits

Find

- Data is easily found
- Information is organized in the way people think about it
- There is an integrated view of information

Relate

- Discovery of data is easy
- Data is smarter and related together

Act

- Preferences for how information is handled
- Work anywhere, anytime, anyplace

Developer Benefits

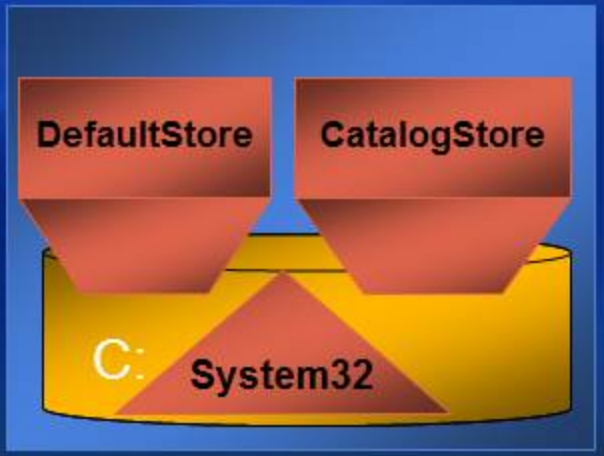
- New Filesystem Capabilities
  - Current applications continue to work on Longhorn, and benefit from new UX capabilities
  - Metadata and relationships on most file formats
  - Metadata handlers for custom file formats
- New Windows primitives
  - New things in the OS to integrate with
  - Extensible to include additional data
- New data subsystem capabilities
  - Smart connected applications
  - Data Sharing

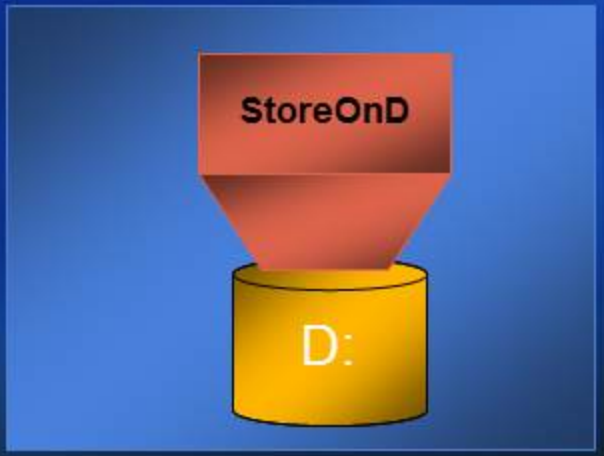
“Marriage”

A filesystem that co-exists with and leverages the best of NTFS

FS Overview
User view – store

- Top-level container for WinFS items
  - More than one store per NTFS volume
  - Stores for system volume created by default
- Integrated into Windows Disk Management and PnP/Power management

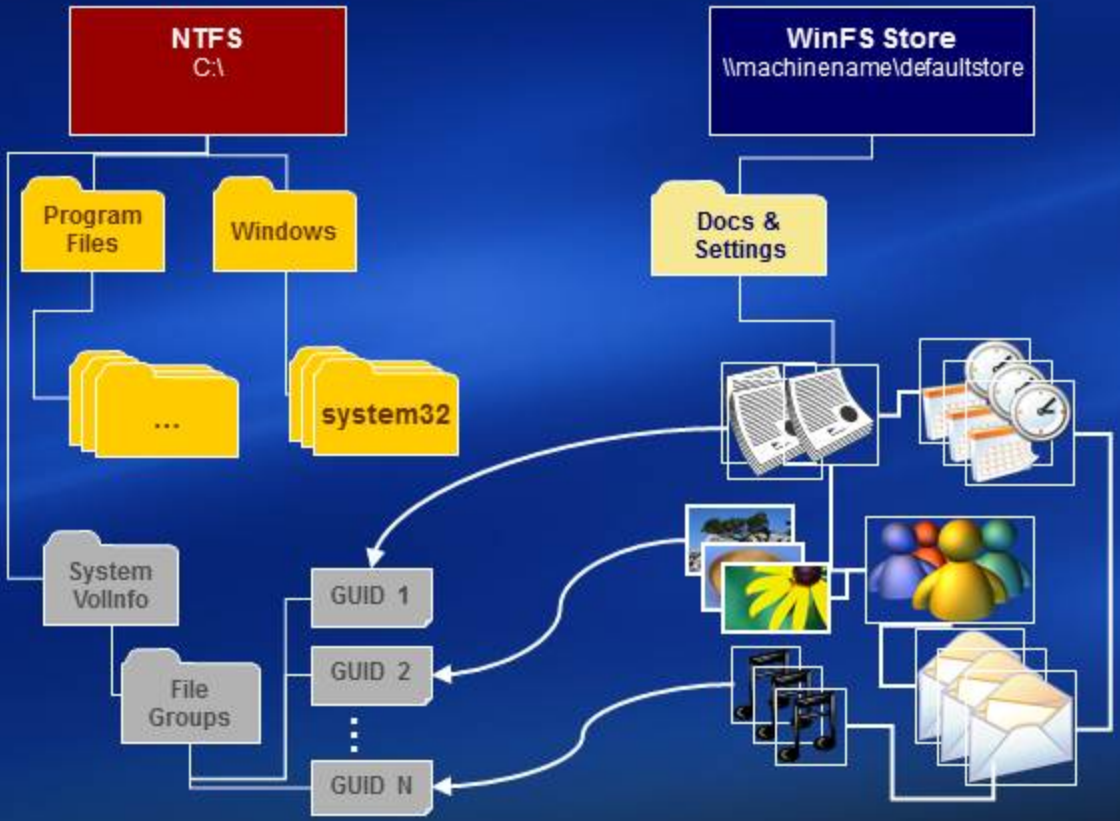




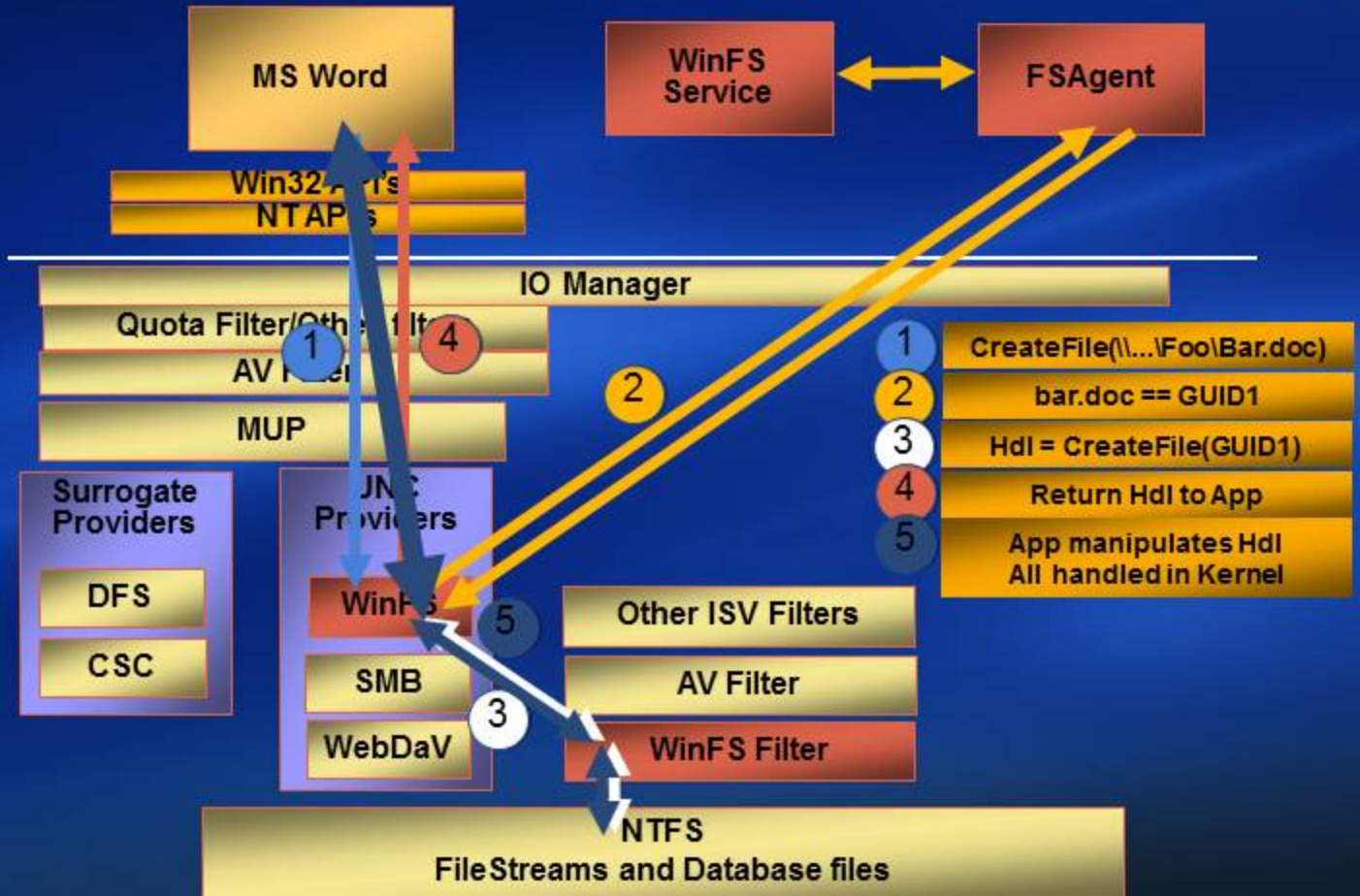
FS Overview
What moves into WinFS

- Targeted user content
  - Fresh install will host Documents, Pictures, Videos in DefaultStore
  - Upgrade will migrate them into DefaultStore
- Not an NTFS replacement
  - Not for Windows directory or Program Files
- Tool will be provided to migrate non-standard directories
  - User content outside of MyDocuments

Leveraging NTFS



FS High Level Architecture



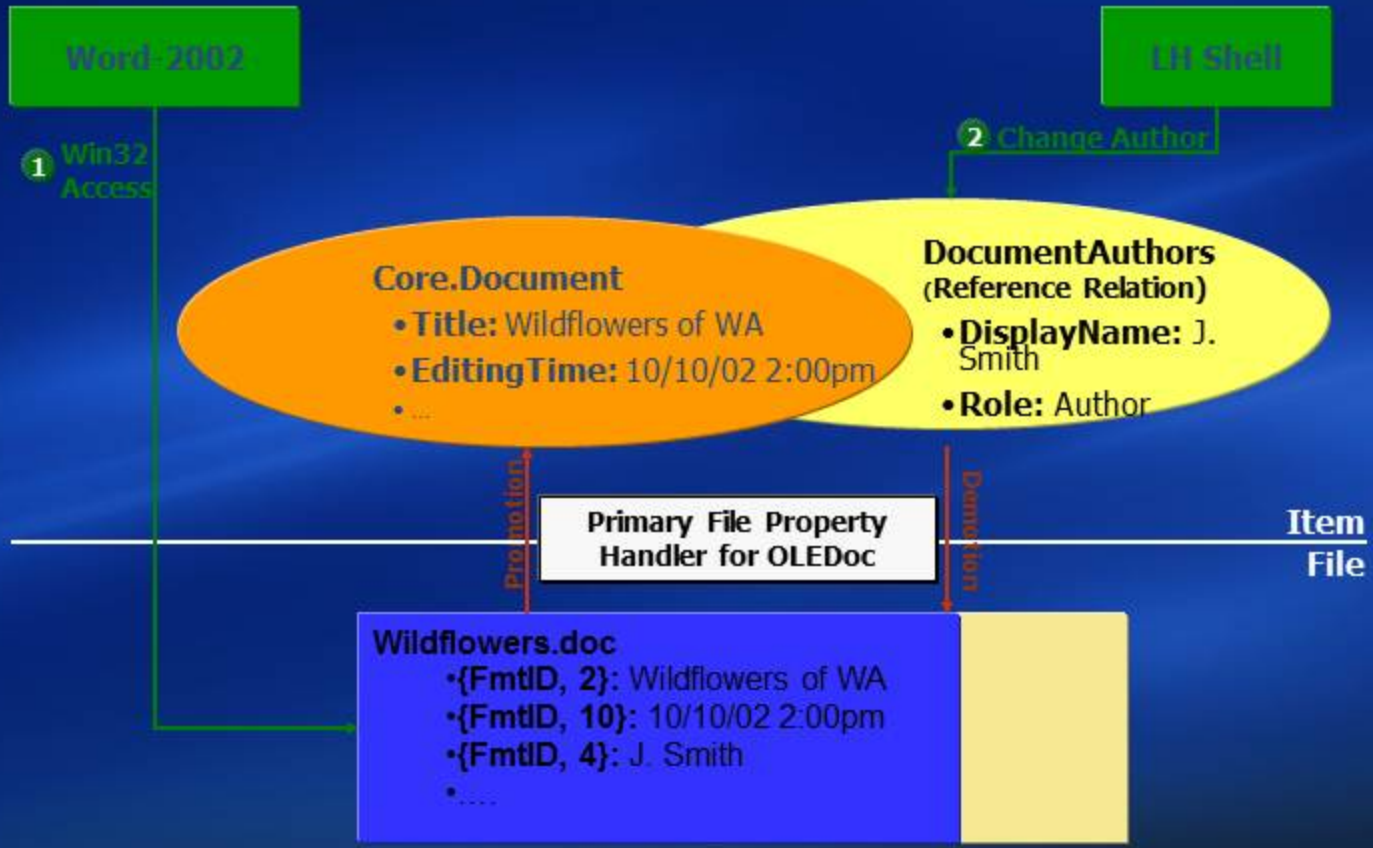
“Freedom”

Surface metadata into platform, that was previously locked inside files

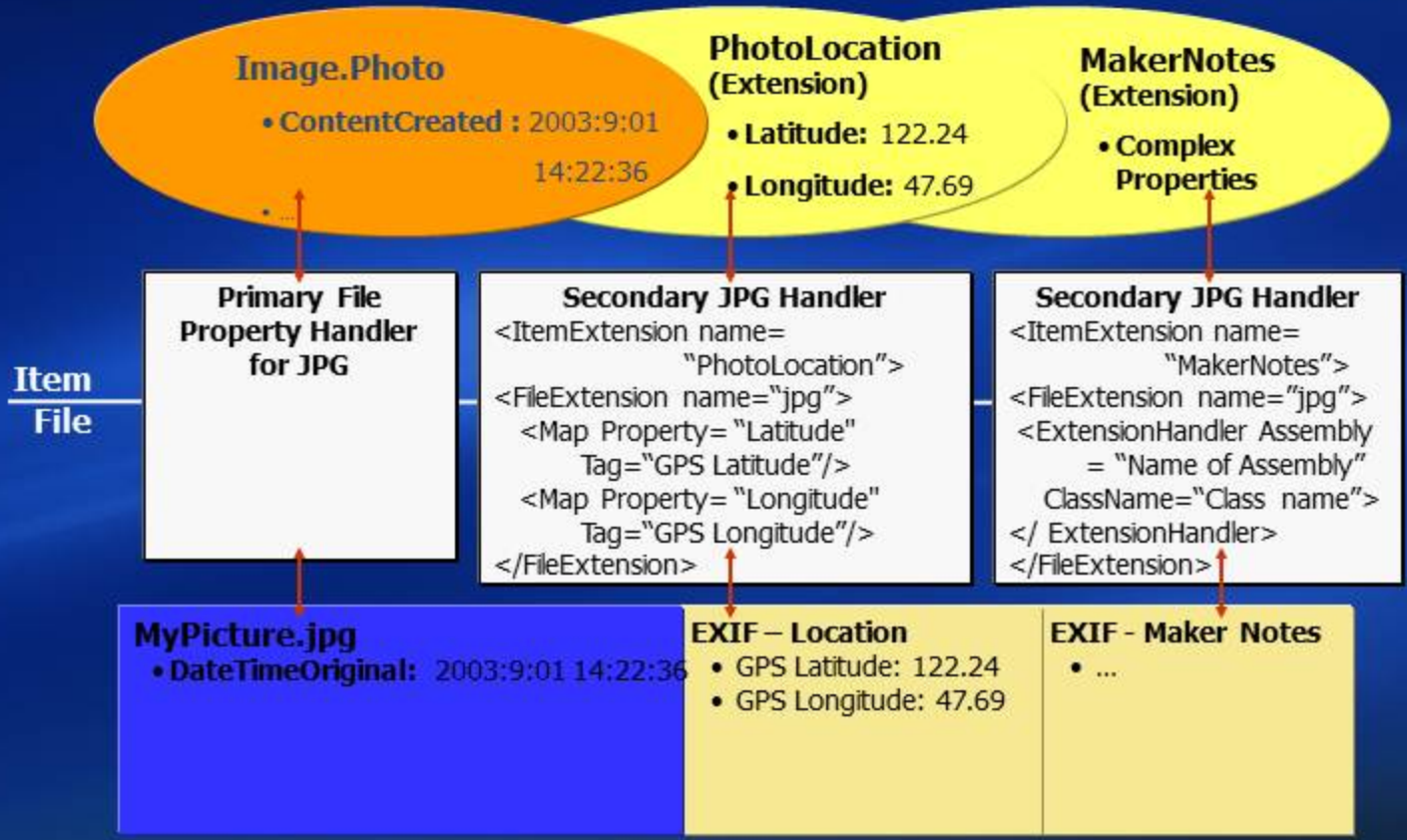
Metadata Handlers
Motivation

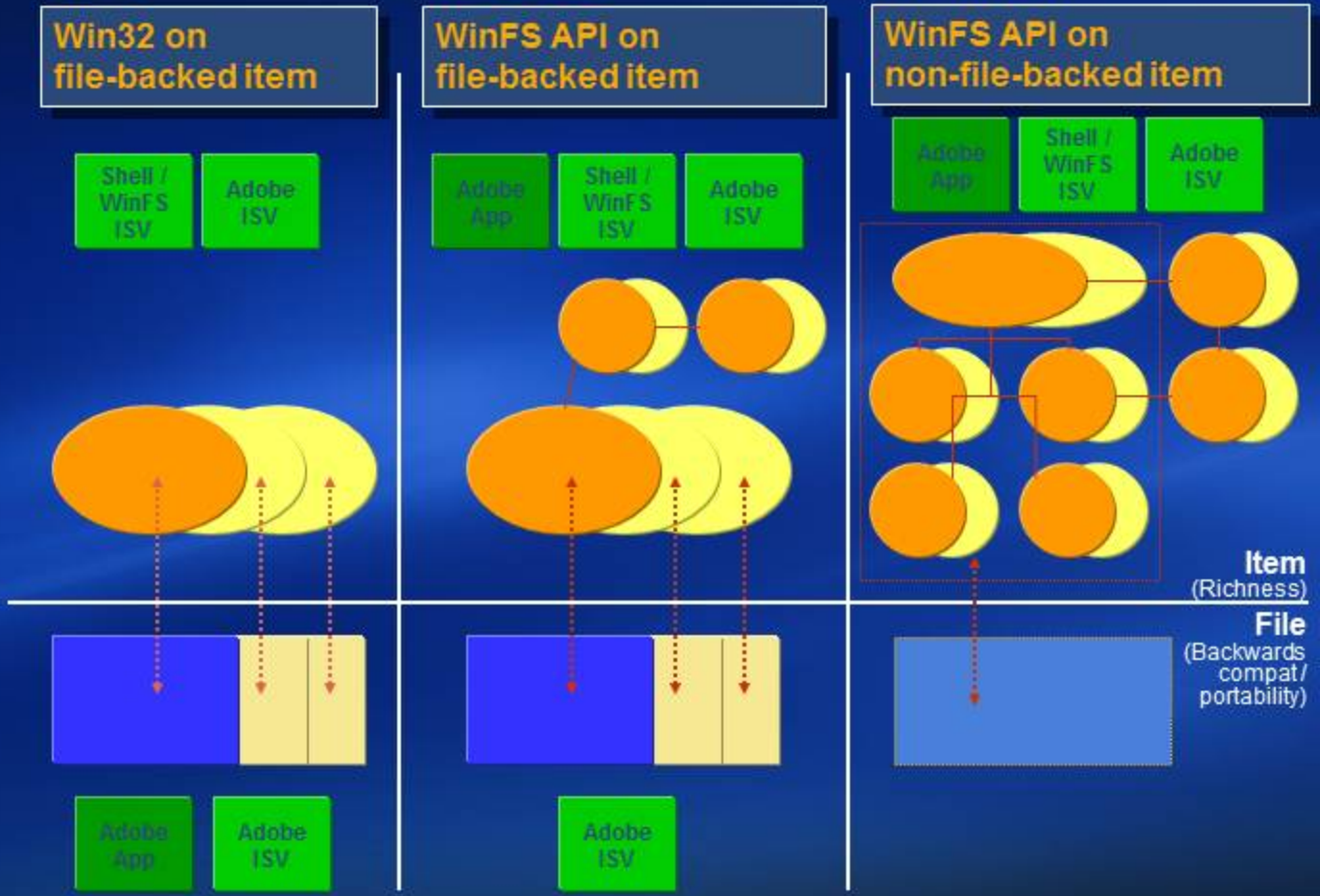
- Promotion
  - End-users don't need to re-tag their content
  - with metadata
    - WinFS automatically pulls it out of files
  - Existing applications continue to write to files
    - Appropriate metadata surfaces in WinFS items
- Demotion
  - WinFS apps use one API to write pure WinFS and file-backed items
    - WinFS demotes appropriate properties to files
  - Allows interop between legacy and new applications
  - Provides fidelity of metadata through moves/copies

Mechanics of Promotion/Demotion



Extensibility




Win32 on file-backed item
WinFS API on file-backed item
WinFS API on non-file-backed item
Shell / WinFS ISV
Adobe ISV
Adobe App
Shell / WinFS ISV
Adobe ISV
Adobe App
Shell / WinFS ISV
Adobe ISV
Item
(Richness)
File
(Backwards compat / portability)
Adobe App
Adobe ISV
Adobe ISV

“Safety”

First class security features keep your data safe

WinFS Security
Core principles

- Compatible with Windows Security Model
  - Authentication, Authorization, Auditing
- Familiar GUI/tools/API's to administer security
  - Maintain Win32 security administration API support
- Improve upon NTFS' security model issues:
  - Security policy is difficult to visualize at a volume level
  - Security policy changes require updates to each file
    - Inefficient even in presence of single-instancing
  - Semantics in presence of hard links are problematic

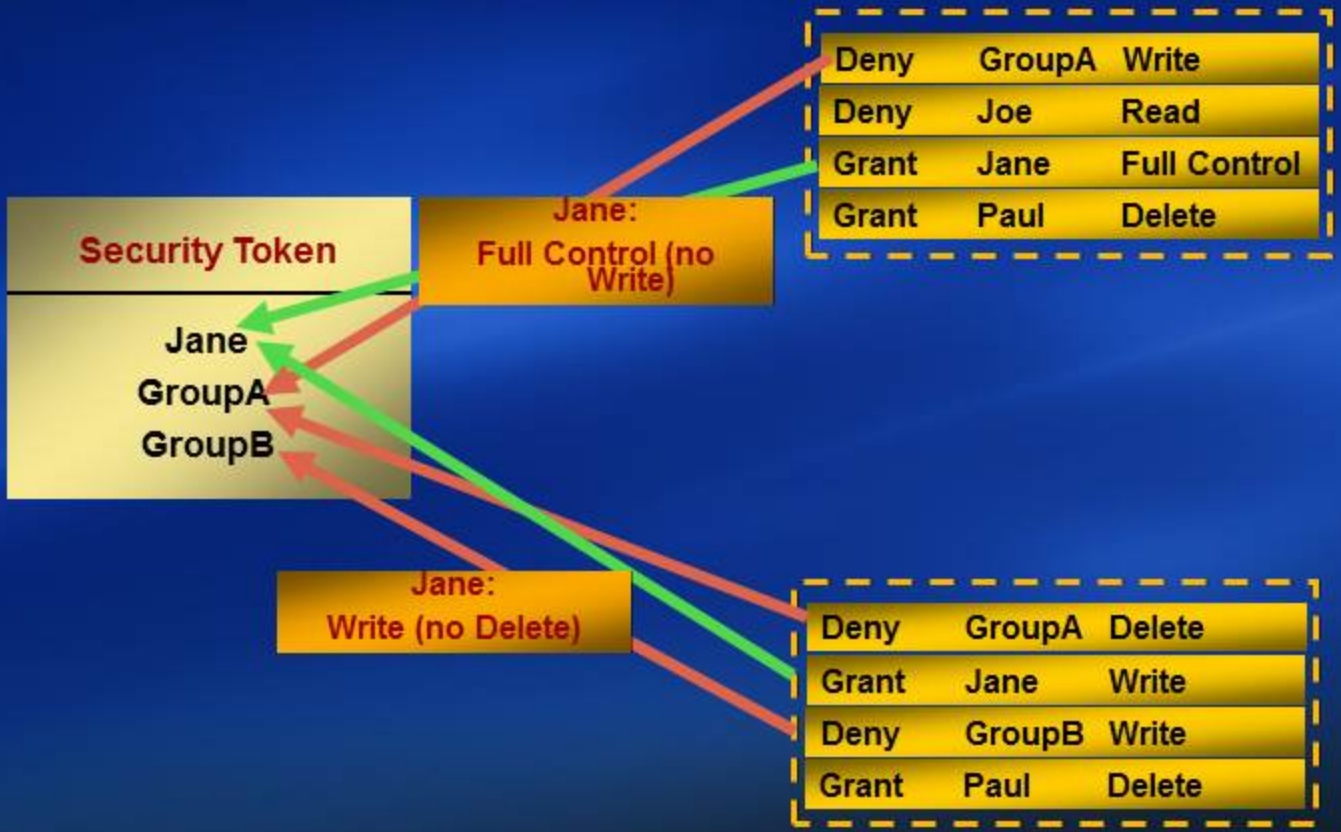
WinFS Security
Authentication

- Directly leverages Windows authentication
- Every operation to WinFS in context of a Security Principal
- Security Principal is represented a Security Token
  - SID, Group membership (SIDs)

WinFS Security
Authorization

- Item is granularity of security policy control
- Namespace hierarchy is mechanism for scalable security administration
- Every item has a Security Descriptor (SD)
  - Discretionary ACL (DACL)
    - Who can (not) access a secure object
  - System ACL (SACL)
    - What system should audit
- Access Control List (ACL)
  - A collection of ACEs
- Access Control Entry (ACE)
  - Grant/Deny, SID, Rights

WinFS Security
Access Check in Action

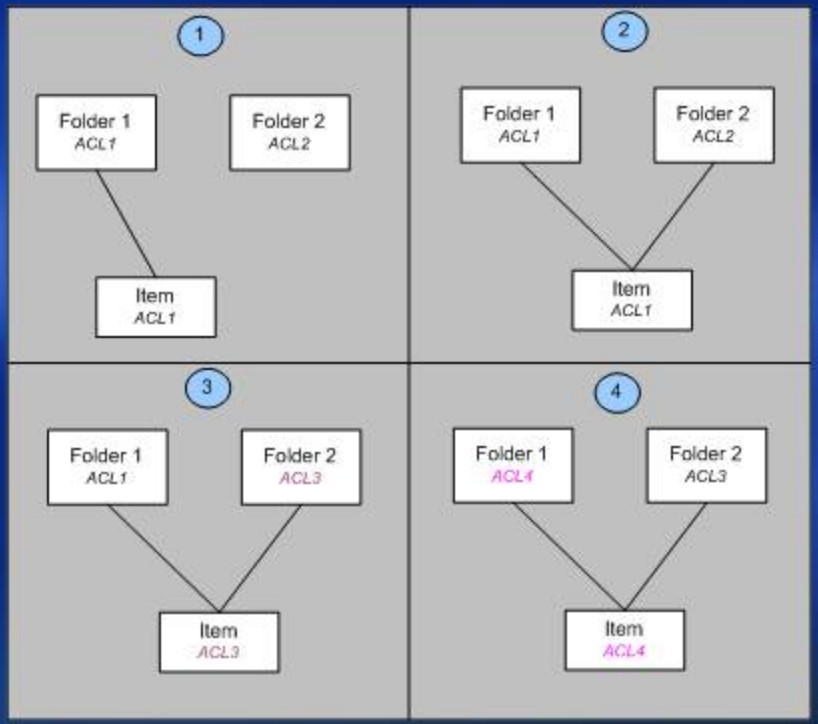


Improvements on NTFS

- ACL's are single instanced based on the sub-tree they apply to
  - Different from current NTFS single instancing
- Each identically protected sub-tree is a unit of administration
  - Better visualization of security policies
- Allows for efficient query evaluation by scoping down to relevant ACL's
- Allows for faster propagation of policies into sub-trees

Improvements on NTFS
Inheritance over Hard links

- NTFS implements "Last-writer-wins"
- Defeats sub-tree based single instancing as way to visualize security



Improvements on NTFS
Inheritance over Holding Links

- Each path contributes a set of inheritable ACE's
- AccessCheck on each combination of Explicit and Inheritable ACE's
  - Grant if at least one Grant and no Deny on any path
- Inheritance rules enforced in API's
  - Not allowed to override Inherited ACE's at Item

Resources/Q/A

- http://desktop – Latest Longhorn Lab06 builds
  - Install a build and see WinFS in action!
- http://winfs – WinFS Team portal
- http://longhorn – Longhorn Client portal
- winfsq@microsoft.com – WinFS Questions alias